**Міністерство освіти і науки України**

**Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”**

**Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії**

Звіт

з лабораторної роботи №3 з дисципліни

“Бази даних”

**“Побудова простих запитів”**

Варіант 32

ВиконалаІП-21 Скрипець Ольга Олександрівна

ПеревірилаМарченко О. І.

**Київ 2023**

**ЛАБОРАТОРНА РОБОТА №3**

**Побудова простих запитів**

**Мета роботи:**

− Вивчити команди DML, котрі використовуються в реляційних СУБД, для вибірки даних з таблиць

− Вивчити команди SQL для створення запитів з використанням підзапитів та з’єднань

− Навчитись створювати запити згідно їх словесного опису

**Постановка задачі лабораторної роботи № 3**

При виконанні лабораторної роботи необхідно виконати наступні дії:

1. Створити запити для вибірки даних з використанням (разом 9 запитів):
2. Найпростіших умов та операторів порівняння
3. Умов з використанням логічних операторів AND, OR та NOT та їх комбінацій.
4. З використанням виразів над стовпцями, як в якості новостворених стовпців, так і умовах
5. Використання операторів:
6. Приналежності множині
7. Приналежності діапазону
8. Відповідності шаблону
9. Перевірка на невизначене значення
10. Створити запити з використанням підзапитів та з’єднань (разом 11 запитів) (в запитах повинні використовуватись 2 та більше таблиць):
    1. Використання підзапитів в рядку вибірки полів (у секції select) та вибірки з таблиць (у секції from)
    2. Використання підзапитів в умовах з конструкціями EXISTS, IN
    3. Декартовий добуток
    4. З’єднання декількох таблиць за рівністю
    5. З’єднання декількох таблиць за рівністю та умовою відбору
    6. Внутрішнього з’єднання
    7. Лівого зовнішнього з’єднання
    8. Правого зовнішнього з’єднання
    9. Об’єднання запитів
11. До кожного запиту з п.1 та 2 навести їх словесний (сутнісний) опис
12. Створити запити за словесним описом, наведеним в завданні згідно варіанту.

**Варіант 32:**

**32) “** Тролейбусне депо **“**

Програмне забезпечення «Тролейбусне депо» . База даних містить інформацію: відомості про водіїв (табельний номер; ПІБ ; дата , час початку і закінчення роботи на маршруті), відомості про тролейбуси (номер ; марка ; кількість пасажирських місць; дата останнього технічного огляду), відомості про маршрути (номер ; час початку і кінця роботи тролейбусів ; початковий пункт ; кінцевий пункт ; список зупинок ; тривалість маршруту). Кожен водій працює на одному тролейбусі і на одному маршруті . На одному маршруті працює кілька водіїв. Кількість пасажирських місць залежить тільки від марки тролейбуса.

**Запити:**

а) Визначить тролейбуси марки «Богдан», котрі працюють на маршрутах, в переліку зупинок яких є зупинка «Політехнічний інститут».

б) Визначить тролейбусні маршрути, за допомогою яких можна дістатися від зупинки «Житомирська» зупинки «Політехнічний інститут» як з пересадкою, так і напряму.

**Виконання:**

* + 1. **Створити запити для вибірки даних з використанням (разом 9 запитів):**

1. **Найпростіших умов та операторів порівняння**

SELECT \* FROM trolleybus WHERE number\_of\_passengers > 25;

1. **Умов з використанням логічних операторів AND, OR та NOT та їх комбінацій.**

SELECT \* FROM trolleybus WHERE number\_of\_passengers > 25 AND wifi\_available = TRUE;

SELECT \* FROM trolleybus WHERE number\_of\_passengers > 30 OR car\_brand\_name = 'Nissan';

SELECT \* FROM trolleybus WHERE NOT (number\_of\_passengers < 20 OR troll\_number = 'T11');

1. **З використанням виразів над стовпцями, як в якості новостворених стовпців, так і умовах**

SELECT troll\_number, car\_brand\_name, number\_of\_passengers,

CASE WHEN wifi\_available THEN 'Available' ELSE 'Not Available' END AS wifi\_status

FROM trolleybus;

1. **Використання операторів:**
2. **Приналежності множині**

SELECT \* FROM schedule WHERE weekday IN ('Monday', 'Wednesday', 'Friday');

1. **Приналежності діапазону**

SELECT \* FROM trolleybus WHERE date\_last\_tech\_insp BETWEEN '2020-11-20' AND '2021-07-20';

1. **Відповідності шаблону**

SELECT \* FROM bus\_stop WHERE bus\_stop\_name LIKE '%View%';

1. **Перевірка на невизначене значення**

SELECT \* FROM trolleybus WHERE date\_last\_tech\_insp IS NULL;

* + 1. **Створити запити з використанням підзапитів та з’єднань (разом 11 запитів) (в запитах повинні використовуватись 2 та більше таблиць):**
  1. **Використання підзапитів в рядку вибірки полів (у секції select) та вибірки з таблиць (у секції from)**

SELECT troll\_number, car\_brand\_name,

(SELECT CASE WHEN car\_brand\_name LIKE '%B%' THEN 'Yes' ELSE 'No' END) AS has\_B\_in\_name

FROM trolleybus;

* 1. **Використання підзапитів в умовах з конструкціями EXISTS, IN**

SELECT \*

FROM route

WHERE EXISTS (

SELECT 1

FROM bus\_stop\_list

WHERE route.list\_of\_bus\_stopID = bus\_stop\_list.list\_of\_bus\_stopID

AND bus\_stop\_list.bus\_stopID = (SELECT bus\_stopID FROM bus\_stop WHERE bus\_stop\_name = 'City Square')

);

SELECT \*

FROM route

WHERE route.list\_of\_bus\_stopID IN (

SELECT list\_of\_bus\_stopID

FROM bus\_stop\_list

WHERE bus\_stopID = (SELECT bus\_stopID FROM bus\_stop WHERE bus\_stop\_name = 'City Square')

);

* 1. **Декартовий добуток**

SELECT troll\_number, route\_number FROM trolleybus, route;

* 1. **З’єднання декількох таблиць за рівністю**

SELECT t.troll\_number, t.car\_brand\_name, c.technical\_characteristics

FROM trolleybus t

JOIN car\_brand c ON t.car\_brand\_name = c.car\_brand\_name;

* 1. **З’єднання декількох таблиць за рівністю та умовою відбору**

SELECT t.troll\_number, t.car\_brand\_name, c.technical\_characteristics

FROM trolleybus t

JOIN car\_brand c ON t.car\_brand\_name = c.car\_brand\_name

WHERE t.number\_of\_passengers > 25;

* 1. **Внутрішнього з’єднання**

SELECT d.DriverID, d.last\_name, d.first\_name, s.weekday

FROM driver d

JOIN schedule s ON d.scheduleID = s.scheduleID;

* 1. **Лівого зовнішнього з’єднання**

SELECT d.DriverID, d.last\_name, d.first\_name, s.weekday

FROM driver d

LEFT JOIN schedule s ON d.scheduleID = s.scheduleID;

* 1. **Правого зовнішнього з’єднання**

SELECT d.DriverID, d.last\_name, d.first\_name, s.weekday

FROM driver d

RIGHT JOIN schedule s ON d.scheduleID = s.scheduleID;

* 1. **Об’єднання запитів**

SELECT start\_work, end\_work FROM schedule

UNION

SELECT last\_name, first\_name FROM driver;

1. **До кожного запиту з п.1 та 2 навести їх словесний (сутнісний) опис**

Пункт 1

a. Найпростіші умови та оператори порівняння: Вивести всі дані з таблиці trolleybus, де кількість пасажирів більше 25.

b. Умови з використанням логічних операторів AND, OR та NOT та їх комбінацій:

* Вивести всі дані з таблиці trolleybus, де кількість пасажирів більше 25 та є доступ до Wi-Fi.
* Вивести всі дані з таблиці trolleybus, де кількість пасажирів більше 30 або марка автобуса - 'Nissan'.
* Вивести всі дані з таблиці trolleybus, де кількість пасажирів не менше 20 і номер тролейбуса не дорівнює 'T11'.

c. Вирази над стовпцями, як в якості новостворених стовпців, так і умовах:

* Вивести номер тролейбусу, марку, кількість пасажирів та статус Wi-Fi (якщо доступно).

d. Використання операторів:

* Приналежність множині: Вивести всі дані з таблиці schedule для понеділка, середи та п'ятниці.
* Приналежність діапазону: Вивести всі дані з таблиці trolleybus, де дата останнього технічного огляду знаходиться в діапазоні від '2020-11-20' до '2021-07-20'.
* Відповідність шаблону: Вивести всі дані з таблиці bus\_stop, де назва зупинки містить 'View'.
* Перевірка на невизначене значення: Вивести всі дані з таблиці trolleybus, де дата останнього технічного огляду є невизначеною.

Пункт 2

a. Використання підзапитів в рядку вибірки полів та вибірки з таблиць:

* Вивести номер тролейбусу, марку, та позначити, чи має марка букву 'B' в назві.

b. Використання підзапитів в умовах з конструкціями EXISTS, IN:

* Вивести всі дані з таблиці route, де існує зупинка з назвою 'City Square' в списку зупинок маршруту.
* Вивести всі дані з таблиці route, де ідентифікатор списку зупинок маршруту знаходиться в списку списків зупинок для зупинки з назвою 'City Square'.

c. Декартовий добуток:

* Вивести всі можливі комбінації номерів тролейбусів та маршрутів без будь-яких умов обрання.

d. З'єднання декількох таблиць за рівністю:

* Вивести номер тролейбусу, марку та технічні характеристики для всіх тролейбусів разом з відповідними характеристиками автомобільних марок.

e. З'єднання декількох таблиць за рівністю та умовою відбору:

* Вивести номер тролейбусу, марку та технічні характеристики для всіх тролейбусів, де кількість пасажирів більше 25.

f. Внутрішнє з'єднання:

* Вивести ідентифікатор водія, прізвище, ім'я водія та день тижня, який відповідає їхньому графіку роботи.

g. Ліве зовнішнє з'єднання:

* Вивести ідентифікатор водія, прізвище, ім'я водія та день тижня, включаючи ті випадки, де графік роботи не визначений.

h. Праве зовнішнє з'єднання:

* Вивести ідентифікатор водія, прізвище, ім'я водія та день тижня, включаючи ті випадки, де графік роботи водія не визначений.

i. Об'єднання запитів:

* Вивести час початку та кінця робочого дня з графіку роботи та прізвище та ім'я водія з таблиці водіїв у вигляді одного набору даних.

1. **Створити запити за словесним описом, наведеним в завданні згідно варіанту.**

**а) Визначить тролейбуси марки «Богдан», котрі працюють на маршрутах, в переліку зупинок яких є зупинка «Політехнічний інститут».**

SELECT DISTINCT t.troll\_number, cb.car\_brand\_name

FROM driver d

JOIN trolleybus t ON d.troll\_number = t.troll\_number

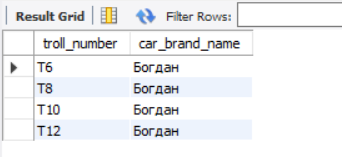
JOIN car\_brand cb ON t.car\_brandID = cb.car\_brandID

JOIN route r ON d.route\_number = r.route\_number

JOIN bus\_stop\_list bsl ON r.route\_number = bsl.route\_number

JOIN bus\_stop bs ON bsl.bus\_stopID = bs.bus\_stopID

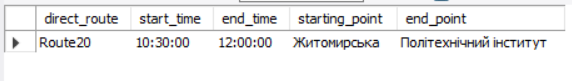
WHERE cb.car\_brand\_name = 'Богдан' AND bs.bus\_stop\_name = 'Політехнічний інститут';

****

**б) Визначить тролейбусні маршрути, за допомогою яких можна дістатися від зупинки «Житомирська» зупинки «Політехнічний інститут» як з пересадкою, так і напряму.**

SELECT DISTINCT r.route\_number AS direct\_route, r.start\_time, r.end\_time, r.starting\_point, r.end\_point FROM route r

WHERE r.starting\_point = 'Житомирська' AND r.end\_point = 'Політехнічний інститут';

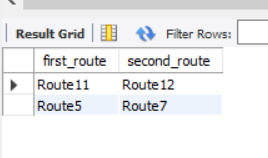
****

SELECT r1.route\_number AS 'first\_route', r2.route\_number AS 'second\_route'

FROM route r1

JOIN route r2 ON r1.end\_point = r2.starting\_point

WHERE r1.starting\_point = 'Житомирська' AND r2.end\_point = 'Політехнічний інститут';

****

**ВИСНОВОК**

В цій лабораторній роботі було вивчено та застосовано різноманітні команди мови SQL для вибірки даних з реляційних таблиць у базі даних. Були використані різні умови та оператори порівняння для отримання конкретних результатів з таблиць.

Також було ознайомлено з використанням підзапитів та з'єднань для складніших та більш гнучких операцій з даними. Підзапити використовувались для вибірки полів у секції SELECT, а також для умов у рядках вибірки та умовах EXISTS, IN. З'єднання таблиць було використано для об'єднання даних з різних таблиць за певними умовами.

Здійснені запити відповідали вимогам завдання та дозволили отримати необхідну інформацію з бази даних. Словесний опис кожного запиту давав зрозуміле уявлення про його функціональність та мету.

Отже, виконання цієї лабораторної роботи допомогло закріпити та розширити знання з мови SQL, розвинути навички написання складних запитів для отримання потрібної інформації з баз даних.